

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.:	10/623,406	§	Examiner:	Lu, Charles Edward
Filed:	July 18, 2003	§	Group/Art Unit:	2161
Inventors:	HanCheng Hsiung Xinyi David Lai	§ § § § §	Atty. Dkt. No:	5760-12400
Title:	REFRESH MECHANISM FOR DATABASES	§ § §		

APPEAL BRIEF

Mail Stop Appeal Brief - Patents

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

Further to the Notice of Appeal mailed May 28, 2008, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

The subject application is owned by VERITAS Operating Corporation, a wholly owned subsidiary of Symantec Corporation.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of related appeals or interferences for the present case.

III. STATUS OF CLAIMS

Claims 1-20 are pending in the case. All of the pending claims stand rejected and are the subject of this appeal. A copy of the claims, as incorporating entered amendments and as on appeal, is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

No amendments have been filed subsequent to the rejection in the Final Office Action of February 28, 2008. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The present application relates to loading new production data into a database using a database clone in order to provide minimally interrupted production database services.

Independent claim 1 recites a system which includes one or more hosts which are configured to implement a production database and a refresh mechanism. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026]-[0029], [0034], [0041]-[0043], [0047], [0055], [0056]; Figures 1, 3, and 4.* The refresh mechanism is configured to generate a storage checkpoint of file system data of the production database. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0025], [0030], [0034], [0035], [0040], [0046], [0047], [0048], [0051]; Figures 1, 2, and 5.* The refresh mechanism is configured to generate a database clone, wherein data of the database clone comprises data from the storage checkpoint. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0030], [0034]-[0036], [0039], [0048], [0051]; Figures 1-5.* The refresh mechanism is configured to load new data to the database clone, wherein said load updates the storage checkpoint, wherein the new data is new with respect to the production database, and wherein the production database is available for access by users during said load. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0029], [0030], [0034]-[0036], [0039], [0040], [0048], [0051]; Figures 1-5.* The refresh mechanism is configured to, after said load, switch from previous file system data of the production database to the storage checkpoint to be the file system data for the production database. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0027], [0029], [0030], [0034]-[0036], [0040], [0043], [0050], [0051]; Figures 1-4.*

Independent claim 8 recites a system which includes means for generating a storage checkpoint of file system data of a production database. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0025], [0030], [0034], [0035], [0040], [0046], [0047], [0048], [0051]; Figures 1, 2, and 5. See also, e.g., at least, paragraphs [0024], [0034], [0035], [0040]-[0043], [0055], [0056]; Figures 3 and 4.* The system further

includes means for generating a database clone of a production database, wherein data of the database clone comprises data from the storage checkpoint. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0030], [0034]-[0036], [0039], [0048], [0051]; Figures 1-5.* *See also, e.g., at least, paragraphs [0024], [0034], [0035], [0040]-[0043], [0055], [0056]; Figures 3 and 4.* The system includes means for loading new data to the database clone, wherein said loading updates the storage checkpoint, wherein the new data is new with respect to the production database, and wherein the production database is available for access by users during said loading. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0029], [0030], [0034]-[0036], [0039], [0040], [0048], [0051]; Figures 1-5. See also, e.g., at least, paragraphs [0024], [0034], [0035], [0040]-[0043], [0055], [0056]; Figures 3 and 4.* The system further includes means for switching from previous file system data of the production database to the storage checkpoint to be the file system data for the production database, wherein said switching is performed after said loading. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0027], [0029], [0030], [0034]-[0036], [0040], [0043], [0050], [0051]; Figures 1-4.* *See also, e.g., at least, paragraphs [0024], [0034], [0035], [0040]-[0043], [0055], [0056]; Figures 3 and 4.*

Independent claim 9 recites a method for refreshing a production database. The method includes generating a storage checkpoint of file system data of the production database. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0025], [0030], [0034], [0035], [0040], [0046], [0047], [0048], [0051]; Figures 1, 2, and 5.* The method also includes generating a database clone of a production database, wherein data of the database clone comprises the storage checkpoint. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0030], [0034]-[0036], [0039], [0048], [0051]; Figures 1-5.* The method includes loading new data to the database clone, wherein said loading updates the storage checkpoint, wherein the new data is new with respect to the production database, and wherein the production database is available for access by users during said loading. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0029], [0030], [0034]-[0036], [0039], [0040], [0048], [0051]; Figures 1-5.* The method also includes, after said loading, switching from previous file system data of the

production database to the storage checkpoint to be the file system data for the production database. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0027], [0029], [0030], [0034]-[0036], [0040], [0043], [0050], [0051]; Figures 1-4.*

Finally, claim 15 recites a computer-accessible storage medium which comprises program instructions that are executable. *See, e.g., at least paragraphs [0050] and [0051].* The program instructions are configured to implement generating a storage checkpoint of file system data of the production database. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0025], [0030], [0034], [0035], [0040], [0046], [0047], [0048], [0051]; Figures 1, 2, and 5.* The program instructions are configured to implement generating a database clone of a production database, wherein data of the database clone comprises the storage checkpoint. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0030], [0034]-[0036], [0039], [0048], [0051]; Figures 1-5.* The program instructions are configured to implement loading new data to the database clone, wherein said loading updates the storage checkpoint, wherein the new data is new with respect to the production database, and wherein the production database is available for access by users during said loading. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0029], [0030], [0034]-[0036], [0039], [0040], [0048], [0051]; Figures 1-5.* The program instructions are configured to implement, after said loading, switching from previous file system data of the production database to the storage checkpoint to be the file system data for the production database. *See, e.g., at least, paragraphs [0015], [0016], [0024], [0026], [0027], [0029], [0030], [0034]-[0036], [0040], [0043], [0050], [0051]; Figures 1-4.*

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-20 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement.
2. Claims 1-20 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite.
3. Claims 1, 2, 5-10, 13-16 and 19-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kampe et al. (U.S. Publication 2002/0032883) (hereinafter “Kampe”) in view of Raman et al. (U.S. Publication 2003/0217119) (hereinafter “Raman”).
4. Claims 3, 11 and 17 stand rejected as being unpatentable over Kampe in view of Raman and further in view of Ishihara et al. (U.S. Patent 6,636,876) (hereinafter “Ishihara”).
5. Claims 4, 12 and 18 stand rejected as being unpatentable over Kampe in view of Raman and further in view of Appellants’ Admitted Prior Art (AAPA).

VII. ARGUMENT

First Ground of Rejection

Claims 1-20 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. More specifically, the Examiner asserts that the specification does not support “wherein the new data is new with respect to the production database”. Appellants respectfully submit that the specification clearly describes the relationship of the new data with respect to the production database. First, Appellants note that the Background section of the instant specification describes the problems associated with loading new data into production databases (e.g., data warehouses). The specification then goes on to describe how to allow new data to be loaded into the production database without interrupting access to the production database (e.g., by using the claimed methods and systems). One of skill in the art would clearly understand that the term “new data” refers to data that is new with respect to the production database. As one specific example, Appellants note paragraph [0036] which recites in part:

In one embodiment, a database clone may be created from a checkpoint of a production database...New data may be loaded into the database clone...During the refresh process, the production database is accessible by users. In one embodiment, once loading data and restructuring the database as necessary is complete, both the production database and the database clone may be shut down. The checkpoint may then be switched to the primary file system. The database clone may be switched to the production database. The production database may then be restarted. The production database has the latest, updated data. (Emphasis Added)

See also, paragraph [0039]:

In a data warehouse environment, data is not typically modified. Typically, the only space required is for newly-loaded data. In embodiments, a database clone includes pointers that point to the original data, and does not include the original data. The database clone initially includes only metadata that points to where the data is. The new data is loaded into the database clone, and updated in the metadata structure for the file system. Thus, in embodiments, one version of a data warehouse may be updated [the database clone] while actively running a current version [the production database] without duplicating the entire database.

The only data storage needed in the database clone is storage for newly-loaded data. (Emphasis Added)

Thus, the specification clearly describes that a database clone is created and new data is loaded into the database clone as a refresh process of the production database. After the new data is loaded in the database clone and the checkpoint is switched to be the primary file system of the production database, the production database has the latest, updated data. One of skill in the art can easily understand from this description (among numerous others in the specification) as well as the identified problem to be solved that the new data is new with respect to the production database. Furthermore, it is clear that the new data is not simply checkpoint data as asserted by the Examiner (addressed in more detail below).

Second Ground of Rejection

Claims 1-20 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. More specifically, the Examiner asserts that the term “new” is a relative term which renders the claim indefinite. The Examiner further asserts that “new” is not defined by the claim, and the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention. Appellants strongly disagree with each of the Examiner’s assertions. First, the claim clearly defines the relationship of the new data with respect to the data of the production database. More specifically, the data is explicitly defined in the claims as new with respect to data of the production database. Appellants submit that it is clear (especially in light of the specification, similar to arguments above) that new production data is being loaded into the database clone and then the file system data of the production database is switched to the storage checkpoint which was just updated with the new data, thereby refreshing the production database. While new is a relative term, Appellants have provided a standard for ascertaining the requisite degree of newness by defining the relationship of the new data with respect to the production data in the claim. Appellants further submit that one of skill in the art could clearly

understand the meaning of new data from the claimed limitations, especially in light of the present specification.

Third Ground of Rejection

Claims 1, 2, 5-10, 13-16 and 19-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kampe et al. (U.S. Publication 2002/0032883) (hereinafter “Kampe”) in view of Raman et al. (U.S. Publication 2003/0217119) (hereinafter “Raman”). Appellant respectfully traverses this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 6-8, 9, 14, 15, and 20

1. Kampe in view of Raman fails to teach or suggest loading new data to the database clone, wherein said loading updates the storage checkpoint, wherein the new data is new with respect to the production database.

First, Appellants note that Kampe is directed towards cluster replication using checkpoints and makes no mention of databases (the term “database”, which has a specific meaning in the art, is found nowhere in Kampe), does not teach the loading of new data into a clone database, and uses the well known method of providing continuous updates for checkpoints to ensure seamless failover and simply does not relate to the problems addressed or methods described by the present claims. The Examiner relies on Raman for teaching the production database and databases in general; however, Raman teaches the transmission of write commands from a first file system to a copy of the file system to maintain concurrency, and thus does not relate to loading new data to a database clone. Additionally, there is no indication that the checkpoints of Kampe could even be used in such a system and the Examiner’s provided reason to combine the reference is no more than hind sight reasoning (which is emphasized by the Examiner’s own assertions in the present Advisory Action). Further, the Examiner’s current

interpretation of “new data” is clearly devoid of merit and does not take into account the plain meaning of Appellants’ claim. Specific deficiencies in the Examiner’s rejection are addressed below.

With respect to the referenced feature, the Examiner relies on Kampe. First, Appellants note that Kampe teaches a system for maintaining replicas of a primary component on nodes (via checkpointing) for achieving high availability in a cluster (see paragraph [0013]). Thus, Kampe is particularly directed towards achieving high availability of a component (in the case of failover) in a cluster rather than loading new data into a production database. For example, in Kampe, paragraph [0064] (which the Examiner relies on for switching from the previous file system data of the production database to the storage checkpoint to be the file system data for the production database), various failover scenarios are discussed where a primary component PC1 fails and the checkpoint is used to recreate the last consistent state of the primary component (either on the first node or on a second node). Thus, in Kampe, checkpoints are used to maintain the current state (or information for recovery) of a component on the first node, and are not used to load new data (new with respect to the production database) into a production database using a database clone. Said another way, Kampe is particularly devoted to loading current state data to the checkpoint. Thus, the checkpoint data is either old with respect to the current state (i.e., the checkpoint does not have the same data and the data is out of date) or up to date with respect to the current state (i.e., the checkpoint and the component have same data). Under no condition is the checkpoint data ever new with respect to the current state (i.e., not the same data and the checkpoint has new data that is not present in the primary component). Thus, Kampe actually **teaches the opposite** from this limitation as Kampe is devoted to replication of data to a checkpoint rather than loading new data into a database clone to update the production database.

Appellants note that the Examiner specifically cites “(504)” for this limitation without any further explanation. 504 is a block in Figure 4A of Kampe which states “continuously update checkpoint”. Thus, 504 relates to maintaining the checkpoint with respect to the current state of the primary component. As argued above, there is no

indication in this Figure or anywhere else (in Kampe or Raman) which relates to loading new data to the database clone, wherein the new data is new with respect to the production database. Using checkpoints to maintain a current state (or backup of a current state) does not relate to loading new information into a database clone (or checkpoint) which is new with respect to the production database (or, using the Examiner's citation, the primary component of Kampe). Thus, claim 1 relates to loading data, which did not previously exist in the production database, into a database clone / checkpoint, and then using the file system data of the checkpoint as the file system data of the production database (thereby loading the new data into the production database). Neither Kampe nor Raman (singly or in combination) teach these features of claim 1. Instead, as indicated above, Kampe maintains a current state of the primary component for failover purposes and does not load new information with respect to the primary component. As indicated above, Appellants further note that loading new data (as recited in Appellants' claims) into the checkpoints of Kampe would be **counter** to the purpose of high availability in a cluster since the replica / checkpoint would no longer represent the current state of the primary component. Thus, the cited references do not teach or suggest loading new data to the database clone, wherein said load updates the storage checkpoint, wherein the new data is new with respect to the production database.

In response to these arguments, the Examiner cites paragraph [0078] of Kampe and describes various data characteristics of the checkpoint of Kampe (e.g., format, states, control blocks, and attributes). The Examiner then goes on to assert these data characteristics "must be 'loaded' as claimed in order to manage the checkpoint". Appellants respectfully submit that this checkpoint attribute data does not relate to the loading of new data of the claims. More specifically, format data, state data, and control block data clearly do not correspond to "loading new data into the checkpoint" and instead are characteristics of the checkpoint itself. Loading new data into a database is a specific process that is different than storing checkpoint characteristic data, as any one of ordinary skill in the art understands. The claimed limitations include generating a database clone which includes a storage checkpoint, loading new data to the database

clone, which updates a storage checkpoint, which is used after the load as the file system data for the production database. Clearly, loading new data is not simply storing characteristic data of the checkpoint (as asserted by the Examiner), but is instead new data that is loaded in the database and becomes production data. Kampe makes a similar distinction between the characteristic data and the actual checkpointing data (see, e.g., paragraph [0085]). Thus, storing characteristic data is clearly not the same as loading new data into the database clone, where the load updates the storage checkpoint, where the new data is new with respect to the production database, and where after the load, the file system data of the checkpoint is used as the file system data of the production database. Furthermore, there is no indication that the checkpoint characteristic data cited by the Examiner is actually used after failover, but instead is characteristic data of the checkpoint.

The Examiner goes on to assert that “Kampe also meets the claim limitation when interpreting ‘new’ as ‘current’, ‘recent’ or ‘fresh’”. The Examiner then goes on to refer to “continuously updating a checkpoint” and asserts that the continuously updated data “thus recently comes into existence during creation/update in steps 503-505, whereas the data in the primary component has already existed”. Appellants submit that the interpretation of ‘new’ as already existing in the production database is entirely without merit and is inapposite to the plain meaning of the claim. Clearly, if the data is the same data that has already existed in the primary component, it is not new data with respect to the production database (primary component in the Examiner’s citation). Appellants further submit that the previously submitted arguments from above appear to be ignored in these assertions.

In the present Advisory Action, the Examiner responds by asserting:

The word “new” is broadly and reasonably interpreted in this situation as being new at least with respect to time. As seen in the prior action, since the prior art “continuously updates” the checkpoint, the data in the checkpoint is newer in time than the data in the production database at certain times, even if other portions of the data may be the same.

The Examiner's interpretation goes beyond reasonably broad and stands the plain meaning on its head. As already noted above, the data of the checkpoint can never be "new data" with respect to the primary component as it is used to back up the primary component, and is indeed a 'replica' of the primary component. One of skill in the art understands that replicas and backups of data do not include new information or data, and if they did, would render the system described in Kampe inoperable. The fact that the data is copied at a later in point in time does not make that data new – the data is still the same data, and is, at best, current data with respect to the primary component. With even a cursory reading of Appellants' claims or specification, one of skill in the art would never interpret new data as currently interpreted by the Examiner.

2. The Examiner has failed to provide a proper reason to combine to the two references.

In the most recent Office Action, the Examiner asserts it would have been obvious to modify Kampe, "to improve accessibility for file system data, as known to one of ordinary skill in the art and taught by Raman (para. 0049)". The Examiner has not provided any reason to make the specific combination and instead has merely provided a portion of Raman which describes a method for providing copies of consistent file systems with concurrent read-write updating of the file system. There is no indication in this section (or in the Examiner's provided motivation) as to why the maintaining of file systems as taught by Raman applies to the "primary component" of Kampe. More specifically, the cited portion does not relate to a "production database", the accessibility of a production database during the load of new data, or any reason to include such features in Kampe as alleged by the Examiner. Instead, Raman teaches that updates may be made over an IP network, and that concurrent read-only access to the remote copies is available.

In the present Advisory Action, the Examiner asserts "It should be noted that the combination was made so that the production database would be available during the

load". Appellants respectfully submit that making a combination for the sole purpose of teaching a claimed limitation is the definition of hindsight reasoning. Additionally, adding a reference only to add a specific teaching is not a reason to combine the references in the first place. The Examiner reiterates that improving accessibility for a file system data is a motivation; however, this only identifies the presumed result of the combination and does not identify any particular reason for one of skill in the art to combine the references. Thus, the Examiner's provided reason to combine the references is **improper** and based on hindsight reasoning.

Claims 2, 10, and 16

Kampe in view of Raman fails to teach or suggest wherein the refresh mechanism is further configured to perform post-processing on the database clone prior to said switch.

With respect to this feature, the Examiner cites "(502-505)" without any further explanation. The cited portion of Kampe relates to creating a checkpoint and continuously updating the checkpoint in order to backup the component PC1. However, there is no indication of performing post-processing on a database clone of a production database prior to switching the database clone to be the production database. Instead, the cited steps relate to keeping a replica up to date (for failover purposes) via a checkpointing service (see paragraphs [0057]-[0060]).

Claims 5, 13, and 19

Kampe in view of Raman fails to teach or suggest wherein the generated database clone includes references to data in the production database.

With respect to this feature, the Examiner asserts:

Kampe as applied above uses a "checkpoint service" and "replica," which must include references to data in the production database or else the data

could not be restored to the primary component (see above and para. 0065).

Appellant respectfully submits that the Examiner is simply incorrect. The checkpoint service and replica does not necessarily include references to data in the primary component (which is not a production database). For example, it is quite likely that the checkpoint service simply copies the data to the replica, and the replica does not include references to data in the production database. In fact, since Kampe is directed to maintaining these replicas for failover purposes (which is clearly not the same as loading new data into a database clone, as already argued), it would not make sense to keep references to the data that should be backed up. In such a case, it would be very likely that in the event of a failover of the primary component, the replica would not be usable for restoring the primary component. Moreover, the cited portion [0065] (502-505 already addressed above) relates to failover of the primary component PC1 and retrieval of data from the checkpoint and makes no mention of a database clone which includes references to data in the production database. Thus, the cited portion of Kampe fails to teach or suggest this feature of the claims.

Fourth Ground of Rejection

Claims 3, 11 and 17 stand rejected as being unpatentable over Kampe in view of Raman and further in view of Ishihara et al. (U.S. Patent 6,636,876) (hereinafter “Ishihara”). Appellant respectfully traverses this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 3, 11, and 17

1. Kampe in view of Raman and Ishihara fails to teach or suggest stopping the production database prior to said switching; and starting the production database after said switching.

With respect to this feature, the Examiner relies on column 6, lines 27-50 of Ishihara. The cited portion teaches the stopping of a first database and the starting of a second duplicate database in order to perform maintenance on the hardware supporting the first database. Appellants submit that stopping a first database and starting a second database is not the same as stopping the first production database, switching the file system to the storage checkpoint (to be the file system data for the production database) and restarting the same production database. Instead, Ishihara teaches starting a different database.

2. The Examiner fails to provide a proper reason to combine the references.

With respect to the combination of Kampe, Raman, and Ishihara, the Examiner asserts:

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Kampe and Raman, such that “stopping the production database prior to the switch and starting the production database after the switch” is implemented. The motivation would have been to facilitate gracefully switching from one data system to another, as known to one of ordinary skill in the art.

Appellants respectfully submit that the Examiner has not provided any reason to make the specific combination and instead has merely provided a summary of the cited portion of Ishihara, which describes stopping a first database and starting a second duplicate database (with no new data) in order to perform maintenance on the hardware supporting the first database. There is no indication in this section (or in the Examiner’s provided motivation) as to why (or how) stopping a first database and starting a second (duplicate) database applies to the “primary component” of Kampe. More specifically, the cited portion does not relate to stopping a database, switching the storage checkpoint to be the file system data of the production database, and starting the production database, and instead, involves starting a different database. Thus, the cited portion and provided motivation does not give a reason to make the specific combination, and is thus **improper**.

Fifth Ground of Rejection

Claims 4, 12 and 18 stand rejected as being unpatentable over Kampe in view of Raman and further in view of Appellants' Admitted Prior Art (AAPA). Appellant respectfully traverses this rejection for at least the reasons given above regarding claims 1, 9, and 15.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-20 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5760-12400/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: July 28, 2008

VIII. CLAIMS APPENDIX

The following lists claims 1-20, incorporating entered amendments, as on appeal.

1. A system, comprising:

one or more hosts configured to implement:

a production database; and

a refresh mechanism configured to:

generate a storage checkpoint of file system data of the production database;

generate a database clone, wherein data of the database clone comprises data from the storage checkpoint;

load new data to the database clone, wherein said load updates the storage checkpoint, wherein the new data is new with respect to the production database, and wherein the production database is available for access by users during said load; and

after said load, switch from previous file system data of the production database to the storage checkpoint to be the file system data for the production database.

2. The system as recited in claim 1, wherein the refresh mechanism is further configured to perform post-processing on the database clone prior to said switch.

3. The system as recited in claim 1, wherein the refresh mechanism is further configured to:

stop the production database prior to said switch; and

start the production database after said switch.

4. The system as recited in claim 1, wherein the production database is a data warehouse.

5. The system as recited in claim 1, wherein the generated database clone includes references to data in the production database.

6. The system as recited in claim 1, wherein the refresh mechanism is configured to perform said loading of new data to the database clone on a different host machine than a host machine hosting the production database.

7. The system as recited in claim 1, wherein the refresh mechanism is configured to perform said loading of new data to the database clone on a host machine hosting the production database.

8. A system, comprising:

means for generating a storage checkpoint of file system data of a production database;

means for generating a database clone of a production database, wherein data of the database clone comprises data from the storage checkpoint;

means for loading new data to the database clone, wherein said loading updates the storage checkpoint, wherein the new data is new with respect to the

production database, and wherein the production database is available for access by users during said loading; and

means for switching from previous file system data of the production database to the storage checkpoint to be the file system data for the production database, wherein said switching is performed after said loading.

9. A method for refreshing a production database, comprising:

generating a storage checkpoint of file system data of the production database;

generating a database clone of a production database, wherein data of the database clone comprises the storage checkpoint;

loading new data to the database clone, wherein said loading updates the storage checkpoint, wherein the new data is new with respect to the production database, and wherein the production database is available for access by users during said loading; and

after said loading, switching from previous file system data of the production database to the storage checkpoint to be the file system data for the production database.

10. The method as recited in claim 9, further comprising performing post-processing on the database clone prior to said switching.

11. The method as recited in claim 9, further comprising:

stopping the production database prior to said switching; and

starting the production database after said switching.

12. The method as recited in claim 9, wherein the production database is a data warehouse.

13. The method as recited in claim 9, wherein the generated database clone includes references to data in the production database.

14. The method as recited in claim 9, wherein said loading new data to the database clone is performed on a different host machine than a host machine hosting the production database.

15. A computer-accessible storage medium comprising program instructions, wherein the program instructions are configured to implement:

generating a storage checkpoint of file system data of the production database;

generating a database clone of a production database, wherein data of the database clone comprises the storage checkpoint;

loading new data to the database clone, wherein said loading updates the storage checkpoint, wherein the new data is new with respect to the production database, and wherein the production database is available for access by users during said loading; and

after said loading, switching from previous file system data of the production database to the storage checkpoint to be the file system data for the production database.

16. The computer-accessible storage medium as recited in claim 15, wherein the program instructions are further configured to implement performing post-processing on the database clone prior to said switching.

17. The computer-accessible storage medium as recited in claim 15, wherein the program instructions are further configured to implement:

stopping the production database prior to said switching; and

starting the production database after said switching.

18. The computer-accessible storage medium as recited in claim 15, wherein the production database is a data warehouse.

19. The computer-accessible storage medium as recited in claim 15, wherein the generated database clone includes references to data in the production database.

20. The computer-accessible storage medium as recited in claim 15, wherein said loading new data to the database clone is performed on a different host machine than a host machine hosting the production database.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no decisions on any related proceedings.